

# Constructing General Orthogonal Fractional Factorial Split-Plot Designs

Bagus Sartono

Department of Statistics, Bogor Agricultural University, Indonesia

Peter Goos

Faculty of Applied Economics and StatUa Center for Statistics, University of Antwerp, Belgium  
Faculty of Bioscience Engineering and Leuven Statistics Research Centre (LSTAT), KU Leuven, Belgium  
Erasmus School of Economics, Erasmus Universiteit Rotterdam, Netherlands

Eric Schoen

TNO, Zeist, Netherlands

Faculty of Applied Economics, University of Antwerp, Belgium

June 27, 2014

## Abstract

While the orthogonal design of split-plot fractional factorial experiments has received much attention already, there are still major voids in the literature. First, designs with one or more factors acting at more than two levels have not yet been considered. Second, published work on nonregular fractional factorial split-plot designs was either based only on Plackett-Burman designs, or on small nonregular designs with limited numbers of factors. In this paper, we present a novel approach to designing general orthogonal fractional factorial split-plot designs. One key feature of our approach is that it can be used to construct two-level designs as well as designs involving one or more factors with more than two levels. Moreover, the approach can be used to create two-level designs that match or outperform alternative designs in the literature, and to create two-level designs that cannot be constructed using existing methodology. Our new approach involves the use of integer linear programming and mixed integer linear programming, and, for large design problems, it combines integer linear programming with variable neighborhood search. We demonstrate the usefulness of our approach by constructing two-level split-plot designs of 16-96 runs, an 81-run three-level split-plot design, and a 48-run mixed-level split-plot design. Supplementary materials for this paper are available online.

*Keywords:* Integer Linear Programming; Mixed-Level Design; Multi-Level Design; Orthogonal Array; Two-Level Design; Variable Neighborhood Search

# 1 Introduction

In industrial experimentation, statistical designs involving several factors are often not completely randomized. Instead, the randomization is restricted to ensure that some factors are reset less frequently than others. The runs of the resulting restricted-randomization experiment are carried out in groups, and the runs in each group have the same setting for certain factors. This type of experiment is referred to as a split-plot experiment. The groups of runs are called whole plots, and the factors whose levels are fixed within a whole plot are the whole-plot factors. The remaining factors are subplot factors.

There are two main reasons to run an experiment using a split-plot rather than a completely randomized design. First, there may be a few factors whose levels are costly or hard to change. Therefore, some authors use the terms hard-to-change and easy-to-change factors to refer to whole-plot and subplot factors, respectively (see, e.g., Anbari and Lucas, 2008; Jones and Nachtsheim, 2009). A split-plot design may also be appropriate when the process under investigation involves two steps, and different factors are applied in each of the steps. For example, Jones and Goos (2007) discuss a split-plot experiment, which involves the production of batches of polypropylene in the first stage and a gas plasma treatment of individual polypropylene plates in the second stage. In that experiment, whole-plot factors (defining the chemical composition of the polypropylene) were applied to the batches, whereas subplot factors (defining the gas plasma treatment) were applied to individual plates.

In this paper, we construct orthogonal split-plot designs. We focus on fractional factorial split-plot (FFSP) designs because, in most practical situations, the available budget is insufficient to run a full factorial design. Most of the literature on orthogonal split-plot designs deals with the construction of regular two-level FFSP designs, or  $2^{k-p}$  split-plot designs, for estimating main effects and two-factor interactions. The  $2^{k-p}$  designs are called regular because they can be constructed by taking a full factorial two-level design in  $k - p$  basic factors with levels  $-1$  and  $+1$ , and calculating the settings of  $p$  additional factors as the products of the settings of two or more basic factors. This construction results in a defining relation of  $2^p - 1$  words that indicate the factorial effects that are completely aliased with the intercept. In such a regular two-level FFSP design, any two factorial effects are either completely aliased with each other, or orthogonal (Montgomery, 2004).

Huang et al. (1998), Bingham and Sitter (1999), Bingham et al. (2004) and Cheng and Tsai (2009) discuss the construction of regular two-level FFSP designs using the aberration criterion, which is based on the length of the words in the defining relation. Capehart et al. (2011) propose a linear programming approach to construct regular two-level FFSP designs which maximize the number of two-factor interactions that are not aliased with main effects and with other two-factor interactions. Regular two-level FFSP designs have two major limitations. First, the number of

runs and the number of whole plots must be powers of two. In contrast, a nonregular two-level FFSP can be constructed for run sizes that are multiples of four, which allows a larger flexibility in run size. A second limitation of regular designs is that they involve completely aliased effects. Therefore, whenever a main effect is aliased with a two-factor interaction effect, the two effects cannot be included simultaneously in a statistical model. In contrast, nonregular designs usually involve partially aliased main effects and interaction effects, so that it is possible to estimate both effects. Nonregular designs' partial aliasing structure is now being recognized as a desirable feature, since it allows experimenters to estimate many more different models than regular designs' complete aliasing structure (Hamada and Wu, 1992).

The advantages of nonregular designs have sparked interest in the construction of nonregular two-level FFSP designs. Kowalski (2002) constructs 24-run designs starting from a 16-run regular FFSP design with four whole plots of four runs. He then applies a semifolding technique to add eight runs, resulting in a 24-run FFSP design with either four whole plots of six runs or six whole plots of four runs. A major weakness of Kowalski's designs is that they are not orthogonal. As a result, the main-effects model, which is the most commonly used model for screening experiments, cannot be estimated with maximum efficiency using these designs.

This weakness is overcome by Tichon et al. (2012), who focus on orthogonal two-level FFSP designs and create a small catalog of two-level FFSP designs with up to 24 runs and eight factors. The small size of the designs in their catalog is due to the fact that their approach involves a complete enumeration of all non-isomorphic two-level orthogonal designs followed by an evaluation of all  $\binom{k}{w}$  possible ways to arrange a  $k$ -factor design into a split-plot design with  $w$  whole-plot factors and  $s = k - w$  subplot factors. This is computationally very intensive, and indicates that there is a need for methodologies to construct large split-plot designs using small designs as building blocks.

Kulahci and Bisgaard (2005) use such an approach and suggest combining small Plackett-Burman designs with either small regular two-level designs or other small Plackett-Burman designs. They show that their approach is useful for robust parameter experiments, where, assuming that the control factors are whole-plot factors and the noise factors are subplot factors (or vice versa), two-factor interactions involving a whole-plot factor and a subplot factor are of primary concern. The work of Kulahci and Bisgaard has several shortcomings. First, their construction only works for specific numbers of runs within a whole plot. In particular, their method can only be used when the number of runs equals the size of a Plackett-Burman design or half the size of a Plackett-Burman design. Second, their construction relies on the use of Plackett-Burman designs and folded-over Plackett-Burman designs, while there are other nonregular two-level designs with better statistical properties as soon as the number of runs is larger than or equal to 20. Third, the use of fold-over designs limits the number of estimable interaction effects. A fold-over design with  $n$  runs consists of  $n/2$  mirror image pairs. Such a design allows

estimation of no more than  $n/2 - 1$  interaction effects. However, Schoen and Mee (2012) show that there exist resolution-4 designs that are not fold-over designs and that permit estimation of a substantially larger number of interactions than  $n/2 - 1$ .

The first main contribution of this paper is that it presents two general methods for constructing orthogonal two-level FFSP designs that overcome the shortcomings of the approaches of Kowalski (2002), Kulahci and Bisgaard (2005) and Tichon et al. (2012), which we use as benchmarks in this paper and its supplementary materials. We should stress, however, that our newly proposed methods are more general in the sense that they can handle any numbers of whole-plot and subplot factors, as well as any numbers of whole plots and subplots that allow for an orthogonal design. The methods do not use fold-over techniques but exploit existing catalogs of orthogonal nonregular designs, and therefore generally result in larger numbers of estimable effects than the approach of Kulahci and Bisgaard (2005).

The second main contribution of the paper is that the newly proposed methods can be used to construct FFSP designs involving one or more factors that have more than two levels. The paper therefore fills an important void in the literature, where the focus clearly is on pure two-level designs despite the fact that there is a need for mixed-level FFSP designs (where not all factors have the same number of levels) and multi-level FFSP designs (where all factors have more than two levels). For example, Schoen and Wolff (1997) discuss an experiment on DNA amplification involving a split-plot design with 12 whole plots of size four, a three-level and a two-level whole-plot factor, a four-level subplot factor and four two-level subplot factors. They used an ad-hoc design construction approach based on a regular two-level FFSP design with eight whole plots of size four.

The two newly proposed methods in this paper involve a stratum-by-stratum construction of the split-plot design, following the approaches of Schoen (1999) and Trinca and Gilmour (2001). The methods combine a select whole-plot design with a select subplot design. Our first method uses integer linear programming to combine the two subdesigns into a final split-plot design in a single step. Our second method to do so is faster, involves two steps and combines integer linear programming with a fast metaheuristic optimization algorithm. The larger speed of the second method makes it suitable for constructing large split-plot designs. We name our two methods the one-step approach and the two-step approach.

Note that our integer linear programming approach is fundamentally different from that of Capehart et al. (2011), because our approach can handle nonregular as well as regular subdesigns, and two-level designs, multi-level designs and mixed-level designs. While our approach produces a final split-plot design by combining two subdesigns, the approach of Capehart et al. (2011) constructs a split-plot design without consideration of subdesigns. Instead, Capehart et al. (2011) seek the ideal defining relation to construct complete split-plot designs in one step. Obviously, by relying on the concept of a defining relation (which cannot be extended to nonregular designs,

and to multi-level and mixed-level designs), Capehart et al. (2011) limit the scope of the problems they can tackle substantially, as well as the set of split-plot scenarios they can consider.

The rest of the paper is structured as follows. The initial focus of the paper is on two-level FFSP designs. In Section 2, we introduce the notation used. We also discuss the confounding frequency vector (CFV; Deng and Tang, 2002) as a measure of the quality of a completely randomized two-level fractional factorial design, and we propose quality measures for two-level FFSP designs based on the CFV. Next, in Section 3, we present our one-step approach to construct two-level FFSP designs and illustrate the approach's added value by means of several examples. We describe our two-step approach for large designs in Section 4. In Section 5, we discuss the generation of multi-level and mixed-level FFSP designs. Throughout the paper, we compare designs constructed with our newly proposed approaches to alternatives from the literature. We end the paper with a discussion of the strengths and weaknesses of our approach and with suggestions for follow-up work. Supplements to this paper are available online, including various additional illustrations, design tables, a study on computing times, and a study of the sensitivity of our approach to a few tuning parameters.

## 2 Preliminaries

In this section, we introduce the main notation and concepts used in this paper. As we introduce our one-step and two-step approaches using two-level FFSP designs, we use the case of two-level factors as a starting point. In Section 5, we explain how to generalize our approaches to mixed-level and multi-level FFSP designs using a minor modification in the notation and the matrices used in the integer linear programming models.

### 2.1 Notation

We denote the set of treatments of a two-level FFSP design by the  $n \times (w + s)$  matrix  $\mathbf{R}$ , where  $n$  denotes the number of runs,  $w$  represents the number of whole-plot factors, and  $s$  is the number of subplot factors. The number of whole plots in the design is denoted by  $b$ , and the number of runs within every whole plot is  $n/b$ . We denote the  $w$  whole-plot factors by  $\mathcal{W}_1, \mathcal{W}_2, \dots, \mathcal{W}_w$  and the  $s$  subplot factors by  $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_s$ . The factor levels are coded as  $-1$  and  $+1$ . The  $b$  whole-plot treatments are collected in a  $b \times w$  matrix  $\mathbf{M}$ . The  $n$  subplot treatments are given by an  $n \times s$  matrix  $\mathbf{S}$ . We call the matrices  $\mathbf{M}$  and  $\mathbf{S}$  whole-plot designs and subplot designs, respectively. The full design  $\mathbf{R}$ , involving  $w + s$  factors, is obtained by assigning every single row of  $\mathbf{M}$  to exactly  $n/b$  rows of  $\mathbf{S}$ .

To describe the assignment, we use an  $n \times b$  assignment matrix  $\mathbf{B} = [b_{ij}]$ , where  $b_{ij} = 1$  if the  $i$ th row of  $\mathbf{S}$  is assigned to the  $j$ th row of  $\mathbf{M}$ , and  $b_{ij} = 0$  otherwise. In other words, if  $b_{ij} = 1$ , then the  $i$ th subplot treatment is applied to a subplot within the  $j$ th whole plot. There must be

$n/b$  ones in each column of  $\mathbf{B}$ , and exactly one such entry in each row. Defining  $\mathbf{W} = \mathbf{B}\mathbf{M}$ , the final design can be represented by  $\mathbf{R} = [\mathbf{W} \mid \mathbf{S}]$ .

In this paper, we require the FFSP designs to be orthogonal and the subplot factors to exhibit level balance in every whole plot. The level balance ensures that the estimates of the subplot factors' main effects are not affected by differences between whole plots and have maximum precision in case a main-effects model is estimated. To obtain the desired kind of FFSP design,  $\mathbf{M}$  and  $\mathbf{S}$  have to be orthogonal, and the assignment matrix  $\mathbf{B}$  must be chosen so that  $\mathbf{B}^T\mathbf{S} = \mathbf{0}$ . For two-level designs, this is possible only if  $b$  and  $n$  are multiples of four and  $n/b$  is even.

## 2.2 Quality measures for fractional factorial designs

### 2.2.1 Completely randomized designs

When a fractional factorial completely randomized design has resolution  $r$ , at least  $r$  main effects are aliased to some extent with  $(r-1)$ -factor interactions and at least  $\binom{r}{q}$   $q$ -factor interactions are aliased to some extent with  $(r-q)$ -factor interactions, where  $q = 2, \dots, r-2$ . This happens whenever there exist sets of  $r$  different factors for which not every combination of levels appears equally often in the design. One way of quantifying this lack of balance is by calculating the so-called  $J_r$  characteristic for each set of  $r$  different factors. For a given set of  $r$  factors, the  $J_r$  characteristic is calculated by taking the product of the factor levels for each of the design's  $n$  runs, summing the  $n$  products, and taking the sum's absolute value. The absolute value of the correlation between a main-effect contrast vector (or that of a  $q$ -factor interaction) and an  $(r-1)$ -factor interaction contrast vector (or that of an  $(r-q)$ -factor interaction) equals  $J_r/n$ .

For two-level designs of resolution  $r$ , all  $J_r$  characteristics have a value of  $n - \lambda 2^r$ , where  $\lambda$  is a non-negative integer and  $n - \lambda 2^r \geq 0$  (Deng and Tang, 1999). So, for designs with 24 runs and resolution 3 for instance, the possible  $J_3$  values are 24, 16, 8, and 0. Therefore, the absolute correlations between any main-effect contrast vector and any two-factor interaction contrast vector can only take the values 1,  $2/3$ ,  $1/3$ , and 0. For 20-run resolution-3 designs, the possible values of the  $J_3$  characteristics are 20, 12, and 4, corresponding to absolute correlations of 1,  $3/5$ , and  $1/5$  (there are no correlations equal to zero). For 16 runs, the possible values are 16, 8, and 0, corresponding to absolute correlations of 1, 0.5, and 0, respectively.

If we denote the total number of factors in a design by  $k$ , there are  $\binom{k}{r}$  sets of  $r$  different factors, each of which has a certain value for the  $J_r$  characteristic. The frequencies of the different values of the  $J_r$  characteristic ( $n, n - 2^r, n - 2 \times 2^r, \dots$ ) are collected in a vector. That vector is the frequency vector  $F_r$ . It is customary to omit the entry for  $J_r = 0$  from  $F_r$ . It is possible to calculate frequency vectors  $F_{r+1}, F_{r+2}, \dots, F_{w+s}$  based on sets of  $r+1, r+2, \dots, w+s$  factors, respectively, in a way similar to  $F_r$ .

Since orthogonal arrays have at least resolution 3, the first frequency vector to be computed

is generally  $F_3$ . The concatenated vector  $(F_3, F_4, \dots, F_k)$  is the confounding frequency vector. For resolution- $r$  designs, it is preferable to have as few  $J_r$  values equal to  $n$  as possible, and, subject to this, as few values equal to  $n - 2^r$  as possible, and so on. This leads to the concept of minimum  $G$ -aberration designs (Deng and Tang, 2002). A minimum  $G$ -aberration design sequentially minimizes the frequencies in the CFV from left to right. A minimum  $G$ -aberration design therefore minimizes the worst types of aliasing.

When comparing designs, we use the truncated confounding frequency vector  $(F_3, F_4)$  because that vector is usually good enough to distinguish between competing designs, and because the frequency vectors  $F_5, \dots, F_k$  are no longer relevant when the primary interest is in main effects and two-factor interactions and higher-order interactions are assumed negligible. The frequency vector  $F_3$  quantifies the degree of aliasing between main effects and two-factor interactions, whereas  $F_4$  measures the degree of aliasing among two-factor interactions.

### 2.2.2 Split-plot designs

In split-plot designs, there are two types of factors: whole-plot factors and subplot factors. As a result, there are three kinds of two-factor interaction effects: interactions involving two whole-plot factors, interactions involving a whole-plot factor and a subplot factor, and interactions involving two subplot factors. We call these interactions  $WW$ ,  $WS$  and  $SS$  interactions, respectively. When calculating the  $J_3$  characteristics, there are four types of three-factor sets. One type involves three whole-plot factors. Another type involves two whole-plot factors and one subplot factor. The third type involve one whole-plot factor and two subplot factors, and the fourth type involves three subplot factors. We can therefore split the  $F_3$  frequency vector into four separate vectors, one for each type of three-factor set. In a similar fashion, we can split the  $F_4$  frequency vector into five separate vectors. Each of the resulting vectors corresponds to a different type of aliasing:

1. The  $WWW$  vector, obtained from all sets of three whole-plot factors, quantifies the degree of aliasing between the main effects of the whole-plot factors and the  $WW$  interactions;
2. The  $WWS$  vector, obtained from all sets of two whole-plot factors and one subplot factor, quantifies the degree of aliasing (i) between the main effects of the subplot factors and the  $WW$  interactions, as well as (ii) between the main effects of the whole-plot factors and the  $WS$  interactions;
3. The  $WSS$  vector, obtained from all sets of one whole-plot factor and two subplot factors, quantifies the degree of aliasing (i) between the main effects of the whole-plot factors and the  $SS$  interactions, as well as (ii) between the main effects of the subplot factors and the  $WS$  interactions;

4. The  $SSS$  vector, obtained from all sets of three subplot factors, quantifies the degree of aliasing between the main effects of the subplot factors and the  $SS$  interactions;
5. The  $WWWW$  vector quantifies the degree of aliasing between all pairs of  $WW$  interactions;
6. The  $WWWS$  vector quantifies the degree of aliasing between  $WW$  interactions and  $WS$  interactions;
7. The  $WWSS$  vector quantifies the degree of aliasing (i) between  $WW$  interactions and  $SS$  interactions, as well as (ii) between all pairs of  $WS$  interactions;
8. The  $WSSS$  vector quantifies the degree of aliasing between  $WS$  interactions and  $SS$  interactions;
9. The  $SSSS$  vector quantifies the degree of aliasing between all pairs of  $SS$  interactions.

The nine vectors can be grouped into three categories. The first category consists of the  $WWS$  and  $WWWS$  vectors. A non-zero entry in one of these vectors would imply that the main effect of a subplot factor is aliased with  $WW$  interaction effects, and, hence, that the subplot main effects are not orthogonal to the whole plots. Since we impose orthogonality between the subplot factors and the whole plots in this paper, the frequencies in the  $WWS$  and  $WWWS$  vectors are all equal zero for any FFSP design we construct. For this reason, we do not explicitly mention the  $WWS$  and  $WWWS$  vectors in the remainder of this paper.

The second category of confounding frequency vectors includes the  $WWW$ ,  $WWWW$ ,  $SSS$  and  $SSSS$  vectors. These four vectors depend only on either the whole-plot design  $\mathbf{M}$  or the subplot design  $\mathbf{S}$ , and, hence, they are not affected by the assignment matrix  $\mathbf{B}$ .

The vectors in the third category are the  $WSS$ ,  $WWSS$  and  $WSSS$  vectors, which do depend on the assignment matrix  $\mathbf{B}$ . For given choices of  $\mathbf{M}$  and  $\mathbf{S}$ , the challenge is to select  $\mathbf{B}$  so that the resulting FFSP design is orthogonal and the elements of the  $WSS$ ,  $WWSS$  and  $WSSS$  vectors are minimal.

Because they indicate aliasing of main effects with two-factor interactions, non-zero entries in the  $WSS$  vector imply a less desirable type of aliasing than non-zero entries in the  $WWSS$  and  $WSSS$  vectors. We therefore prioritize the minimization of aliasing of the type  $WSS$ . An additional argument to do so is that aliasing of the type  $WSS$  implies that  $SS$  interactions are at least partially aliased with whole-plot main effects, so that information concerning these interactions appears in the whole-plot stratum. This potentially has a detrimental impact of the estimation precision of the  $SS$  interactions.

We also attach a larger importance to the minimization of the  $WWSS$  vector's elements than to the minimization of the  $WSSS$  vector's elements, because aliasing of the type  $WWSS$  implies that  $SS$  interactions are at least partially aliased with  $WW$  interactions. This may also



cause the  $\mathcal{SS}$  interaction effects to be estimated less precisely. Aliasing of the type  $\mathcal{WSSS}$  does not result in information concerning  $\mathcal{SS}$  interactions appearing in the whole-plot stratum.

We quantify the  $\mathcal{WSS}$  aliasing using the matrix product  $\mathbf{D} = \mathbf{Z}^T \mathbf{W}$ , where  $\mathbf{Z}$  is the  $n \times s_2$  matrix containing all  $s_2 = s(s-1)/2$   $\mathcal{SS}$  interaction contrast vectors. The columns of  $\mathbf{Z}$  are obtained by elementwise multiplication of any two different columns of  $\mathbf{S}$ . The entries of  $\mathbf{D}$  are plus or minus the  $J_3$  characteristics corresponding to  $\mathcal{WSS}$  aliasing.

To achieve the smallest degree of aliasing of the type  $\mathcal{WSS}$ , the entries of  $\mathbf{D}$  should be as close to zero as possible. Denoting the absolute value of the entry in the  $i$ th row and  $j$ th column of  $\mathbf{D}$  by  $d_{ij}$ , all  $d_{ij}$  values should be as small as possible. We achieve this by sequentially minimizing the maximum  $d_{ij}$  value,

$$d_0 = \max\{d_{ij}\}, \quad (1)$$

and the sum of all  $d_{ij}$  values,

$$d_1 = \sum_{i=1}^{s_2} \sum_{j=1}^w d_{ij}. \quad (2)$$

By prioritizing the minimization of  $d_0$ , we ensure that none of the  $\mathcal{SS}$  interactions is severely aliased with a whole-plot factor's main effect, and that none of the  $\mathcal{WS}$  interactions is severely aliased with a subplot factor's main effect. By subsequently minimizing  $d_1$ , we ensure that the remaining aliasing of the type  $\mathcal{WSS}$  is as small as possible too. This sequential minimization approach first ensures that as many of the leftmost elements of the  $\mathcal{WSS}$  vector as possible are zero. Next, by minimizing the sum of all  $J_3$  characteristics, it makes the remaining elements of the  $\mathcal{WSS}$  vector as small as possible.

To quantify the aliasing of the types  $\mathcal{WWSS}$  and  $\mathcal{WSSS}$ , we first define the matrices  $\mathbf{E} = \mathbf{Z}^T \mathbf{V}$  and  $\mathbf{F} = \mathbf{Z}^T \mathbf{T}$ . The  $n \times w_2$  matrix  $\mathbf{V}$  contains all  $w_2 = w(w-1)/2$  vectors obtained by elementwise multiplication of two different columns of  $\mathbf{W}$ . Thus,  $\mathbf{V}$  is the matrix of  $\mathcal{WW}$  interaction contrast vectors. The  $n \times ws$  matrix  $\mathbf{T}$  contains the  $\mathcal{WS}$  interaction contrast vectors. It is obtained as  $\mathbf{T} = (\mathbf{1}_w^T \otimes \mathbf{S}) \circ (\mathbf{W} \otimes \mathbf{1}_{n/b}^T)$ , where  $\otimes$  and  $\circ$  are the Kronecker and the Hadamard (elementwise) product operators, respectively. As a result,  $\mathbf{E}$  measures the degree of aliasing of the type  $\mathcal{WWSS}$  and  $\mathbf{F}$  measures the degree of aliasing of the type  $\mathcal{WSSS}$ .

Next, we define  $e_0$  and  $e_1$ , and  $f_0$  and  $f_1$  corresponding to  $\mathbf{E}$  and  $\mathbf{F}$  analogously to  $d_0$  and  $d_1$  in (1) and (2). Finally, by choosing appropriate weights, we prioritize the minimization of  $e_0$  over the minimization of  $e_1$ , and the minimization of  $f_0$  over the minimization of  $f_1$ , to ensure that, first, the most severe aliasing among the two-factor interactions is minimized, and, next, there is as little remaining aliasing as possible.

### 3 One-step approach for small two-level designs

In this section, we outline our one-step approach to generate two-level FFSP designs, which employs integer linear programming (ILP). We discuss two illustrations to compare the performance of the resulting designs with that of alternative designs presented in the literature (two more illustrations are included in Supplementary Sections A1 and A2). We also briefly discuss the computing times and the sensitivity of the results to the weights of  $d_0$ ,  $d_1$ ,  $e_0$ ,  $e_1$ ,  $f_0$  and  $f_1$  (more details are given in Supplementary Sections B and C). While our focus in this section is on scenarios for which benchmarks exist in the literature, we would like to stress that our ILP approach is capable of handling many scenarios on which the literature is silent.

We implemented the one-step approach in SAS and Matlab. From Matlab, we call the solver CPLEX, although some other solvers could be used too. These implementations, along with some instructions concerning their use, can be found in the supplementary materials.

#### 3.1 Integer linear programming model

Recall that our approach starts by determining a design for the whole-plot treatments,  $\mathbf{M}$ , and a design for the subplot treatments,  $\mathbf{S}$ , and that the final split-plot arrangement is of the form  $\mathbf{R} = [\mathbf{W} \mid \mathbf{S}] = [\mathbf{B}\mathbf{M} \mid \mathbf{S}]$ . Given  $\mathbf{M}$  and  $\mathbf{S}$ , the only remaining issue is to find the assignment matrix  $\mathbf{B}$ . This matrix is determined so that the final FFSP design is orthogonal and performs well in terms of the aliasing of the types  $\mathcal{WSS}$ ,  $\mathcal{WWSS}$  and  $\mathcal{WSSS}$ . These three types of aliasing are quantified by the matrices  $\mathbf{D}$ ,  $\mathbf{E}$  and  $\mathbf{F}$ , respectively.

We optimize  $\mathbf{B}$  with integer linear programming (ILP). The objective function we minimize is a weighted sum of the  $d_0$ ,  $d_1$ ,  $e_0$ ,  $e_1$ ,  $f_0$  and  $f_1$  values defined in Section 2.2.2. ILP is a variant of linear programming commonly used for solving linear optimization problems and involving integer decision variables only. We use the following ILP model to construct FFSP designs in our one-step approach:

$$\text{Minimize } c_{d_0}d_0 + c_{d_1}d_1 + c_{e_0}e_0 + c_{e_1}e_1 + c_{f_0}f_0 + c_{f_1}f_1, \quad (3)$$

Subject to

Orthogonality constraints:

$$\mathbf{S}^T \mathbf{B} = \mathbf{0}_{s \times b}, \quad (4)$$

$\mathcal{WSS}$  constraints:

$$\mathbf{z}_i^T \mathbf{w}_j - d_{ij} \leq 0; \quad i = 1, \dots, s_2; j = 1, \dots, w, \quad (5)$$

$$\mathbf{z}_i^T \mathbf{w}_j + d_{ij} \geq 0; \quad i = 1, \dots, s_2; j = 1, \dots, w, \quad (6)$$

$$0 \leq d_{ij} \leq d_0; \quad i = 1, \dots, s_2; j = 1, \dots, w, \quad (7)$$

$$d_1 = \sum_{i=1}^{s_2} \sum_{j=1}^w d_{ij}; \quad (8)$$

$\mathcal{WWS}$  constraints:

$$\mathbf{z}_i^T \mathbf{v}_j - e_{ij} \leq 0; \quad i = 1, \dots, s_2; j = 1, \dots, w_2, \quad (9)$$

$$\mathbf{z}_i^T \mathbf{v}_j + e_{ij} \geq 0; \quad i = 1, \dots, s_2; j = 1, \dots, w_2, \quad (10)$$

$$0 \leq e_{ij} \leq e_0; \quad i = 1, \dots, s_2; j = 1, \dots, w_2, \quad (11)$$

$$e_1 = \sum_{i=1}^{s_2} \sum_{j=1}^{w_2} e_{ij}; \quad (12)$$

$\mathcal{WSS}$  constraints:

$$\mathbf{z}_i^T \mathbf{t}_j - f_{ij} \leq 0; \quad i = 1, \dots, s_2; j = 1, \dots, ws, \quad (13)$$

$$\mathbf{z}_i^T \mathbf{t}_j + f_{ij} \geq 0; \quad i = 1, \dots, s_2; j = 1, \dots, ws, \quad (14)$$

$$0 \leq f_{ij} \leq f_0; \quad i = 1, \dots, s_2; j = 1, \dots, ws, \quad (15)$$

$$f_1 = \sum_{i=1}^{s_2} \sum_{j=1}^{ws} f_{ij}; \quad (16)$$

Technical constraints:

$$\mathbf{1}_n^T \mathbf{B} = \frac{n}{b} \mathbf{1}_b^T, \quad (17)$$

$$\mathbf{B} \mathbf{1}_b = \mathbf{1}_n, \quad (18)$$

$$b_{ij} \in \{0, 1\}; \quad i = 1, \dots, n; j = 1, \dots, b, \quad (19)$$

In this formulation,  $\mathbf{z}_i$ ,  $\mathbf{w}_i$ ,  $\mathbf{v}_i$ , and  $\mathbf{t}_i$  represent the  $i$ th column of the matrices  $\mathbf{Z}$ ,  $\mathbf{W}$ ,  $\mathbf{V}$ , and  $\mathbf{T}$ , respectively,  $\mathbf{0}_{s \times b}$  is an  $(s \times b)$ -dimensional zero matrix, and  $\mathbf{1}_b$  and  $\mathbf{1}_n$  are  $b$ - and  $n$ -dimensional vectors of ones, respectively. Finally,  $b_{ij}$  represents the element of  $\mathbf{B}$  in the  $i$ th row and  $j$ th column.

All constraints as well as the objective function in the above model are linear in the decision variables  $b_{ij}$ ,  $d_{ij}$ ,  $e_{ij}$ ,  $f_{ij}$ ,  $d_0$ ,  $e_0$ ,  $f_0$ ,  $d_1$ ,  $e_1$ , and  $f_1$ . Moreover, the unknown variables are all integer. The entries  $b_{ij}$  of the assignment matrix  $\mathbf{B}$  are decision variables whose values are either 0 or 1. The  $d_{ij}$ ,  $e_{ij}$  and  $f_{ij}$  values result from multiplying the binary matrix  $\mathbf{B}$  and some given integer matrices, so they can only take integer values. As a consequence, the  $d_0$ ,  $d_1$ ,  $e_0$ ,  $e_1$ ,  $f_0$  and  $f_1$  values (which are the maxima and sums of the  $d_{ij}$ ,  $e_{ij}$  and  $f_{ij}$  values) are also integer.

There are five groups of constraints in the ILP model. The orthogonality constraints (4) ensure that the subplot factors are orthogonal to the whole plots, and, hence, that the subplot design and the whole-plot design are combined so that the resulting FFSP is orthogonal.

The next three groups of constraints are alike and consist of auxiliary constraints needed to

define the components of the objective function. For example, the constraints (5)-(8) quantify the aliasing of the type  $\mathcal{WSS}$ . Constraints (5)-(6) and the non-negativity constraint (7) define the  $d_{ij}$  values as the absolute values of the entries of  $\mathbf{D} = \mathbf{Z}^T \mathbf{W}$ : if  $\mathbf{z}_i$  is the  $i$ th column of  $\mathbf{Z}$  and  $\mathbf{w}_j$  is the  $j$ th column of  $\mathbf{W}$ , then  $d_{ij}$  is the absolute value of the element in row  $i$  and column  $j$  of  $\mathbf{D}$ . The types of constraints given in (5)-(6), (9)-(10) and (13)-(14) utilize the standard way of linearizing the absolute value function (which is a non-linear function) in ILP. The constraints  $d_{ij} \leq d_0$ ,  $e_{ij} \leq e_0$  and  $f_{ij} \leq f_0$  in (7), (11) and (15) define  $d_0$ ,  $e_0$  and  $f_0$  as the maximum  $d_{ij}$ ,  $e_{ij}$  and  $f_{ij}$  values. The constraints (8), (12) and (16) define  $d_1$ ,  $e_1$  and  $f_1$  as the sum of the  $d_{ij}$ ,  $e_{ij}$  and  $f_{ij}$  values.

The technical constraints (17)-(19) ensure that the assignment matrix  $\mathbf{B}$  has the desired structure. Constraint (19) states that  $\mathbf{B}$  is a binary matrix. The other constraints state that there should be exactly one unit element in each row and  $n/b$  unit elements in every column.

As explained earlier, we attach the highest priority to the minimization of the  $\mathcal{WSS}$ -type aliasing and the lowest priority to the minimization of the  $\mathcal{WSSS}$ -type aliasing. Therefore, in our approach, the weights of  $d_0$ ,  $d_1$ ,  $e_0$ ,  $e_1$ ,  $f_0$  and  $f_1$  in the objective function satisfy the inequalities  $c_{d_q} > c_{e_q} > c_{f_q}$ , for  $q = 0, 1$ . In addition, we set  $c_{d_0} > c_{d_1}$ ,  $c_{e_0} > c_{e_1}$  and  $c_{f_0} > c_{f_1}$  to ensure that, within each type of aliasing, the maximum degree of aliasing is minimized first. For the examples we describe below, we used  $c_{d_0} = 10^7$ ,  $c_{d_1} = 10^3$ ,  $c_{e_0} = 10^5$ ,  $c_{e_1} = 10^1$ ,  $c_{f_0} = 10^3$ , and  $c_{f_1} = 10^{-1}$ . We study the sensitivity of the designs produced by the one-step approach to the weights of  $d_0$ ,  $d_1$ ,  $e_0$ ,  $e_1$ ,  $f_0$  and  $f_1$  in Supplementary Section C1 and summarize the results in Section 3.3.

## 3.2 Illustrations

We compared 16-run, 24-run and 32-run FFSP designs obtained using the one-step ILP approach with alternatives found in the literature. Table 1 presents an overview of the cases along with references to the articles that describe the benchmarks. In this section, we report the results for the 16-run design in four whole plots of size four and for the 32-run design in eight whole plots of size four. The results for the other two cases are discussed in Supplementary Sections A1 and A2.

Table 1: Examples used to illustrate the one-step ILP approach.

$n$	$b$	$w$	$s$	Benchmark design(s)
16	4	3	5	Bingham and Sitter (1999); Capehart et al. (2011); Tichon et al. (2012)
24	4	2	4	Kowalski (2002); Tichon et al. (2012)
32	8	3	5	Huang et al. (1998); Capehart et al. (2011)
32	8	2	7	Bingham et al. (2004)

### 3.2.1 16-run designs

Our first example is a FFSP design with three whole-plot factors ( $\mathcal{W}_1, \mathcal{W}_2, \mathcal{W}_3$ ), five subplot factors ( $\mathcal{S}_1, \dots, \mathcal{S}_5$ ), and four whole plots of four runs. Bingham and Sitter (1999) provide a regular minimum aberration design for this scenario, while Capehart et al. (2011) provide a regular design that maximizes the number of two-factor interactions that are not aliased with main effects or other two-factor interactions. The former design, labeled **BS16** in this paper, has generators  $\mathcal{W}_3 = \mathcal{W}_1\mathcal{W}_2$ ,  $\mathcal{S}_3 = \mathcal{W}_1\mathcal{S}_1$ ,  $\mathcal{S}_4 = \mathcal{W}_1\mathcal{S}_2$  and  $\mathcal{S}_5 = \mathcal{W}_2\mathcal{S}_1\mathcal{S}_2$ . The latter, labeled **C16**, has generators  $\mathcal{W}_3 = \mathcal{W}_1\mathcal{W}_2$ ,  $\mathcal{S}_3 = \mathcal{W}_1\mathcal{S}_1$ ,  $\mathcal{S}_4 = \mathcal{W}_2\mathcal{S}_1$  and  $\mathcal{S}_5 = \mathcal{W}_3\mathcal{S}_1$ . Another benchmark design is given by Tichon et al. (2012), who explored all possible nonregular FFSP designs. They report a nonregular generalized minimum aberration design for this scenario, which we label **T16**.

Our ILP approach requires the selection of a  $16 \times 5$  orthogonal subplot subdesign **S** for the subplot treatments and a  $4 \times 3$  subdesign **M** for the whole-plot treatments. A natural choice for **S** is the resolution-5 regular half fraction of the  $2^5$  design with defining relation  $I = \mathcal{S}_1\mathcal{S}_2\mathcal{S}_3\mathcal{S}_4\mathcal{S}_5$ . The only design option for the whole-plot treatments is the regular  $2^{3-1}$  design with generator  $\mathcal{W}_3 = \mathcal{W}_1\mathcal{W}_2$ . The FFSP design obtained using the ILP model and labeled **ILP16** is presented in Table 2.

In Table 3, we compare the designs **ILP16**, **BS16**, **C16** and **T16** with respect to the  $F_3$  and  $F_4$  frequency vectors. The table's upper panel shows the results for the  $\mathcal{SSS}$ ,  $\mathcal{WSS}$  and  $\mathcal{WWW}$  vectors, while the lower panel shows the results for the  $\mathcal{SSSS}$ ,  $\mathcal{WSSS}$  and  $\mathcal{WWSS}$  vectors.

The designs **BS16** and **C16** are regular. Factorial effects in regular designs are either completely aliased or orthogonal. For this reason, a regular design can only have  $J_3$  and  $J_4$  characteristics of  $n$  or 0. The second entry in the  $F_3$  and  $F_4$  confounding frequency vectors for these designs, corresponding to a  $J_3$  or  $J_4$  characteristic of 8, is therefore zero. The designs **ILP16** and **T16** have regular subdesigns for both the whole-plot as well as the subplot treatments. However, the presence of  $J_3$  and  $J_4$  characteristics equalling 8 shows that the two combined

Table 2: 16-run FFSP design with 3 whole-plot factors, 5 subplot factors and 4 whole plots of size 4, constructed using the ILP model in Section 3.1 and labeled **ILP16**.

WP	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_3$	SP	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_4$	$\mathcal{S}_5$
1	+	+	+	1	−	−	+	+	−
				2	−	+	−	−	+
				3	+	−	+	−	−
				4	+	+	−	+	+
2	+	−	−	1	−	−	−	−	−
				2	−	−	−	+	+
				3	+	+	+	−	+
				4	+	+	+	+	−
3	−	+	−	1	−	−	+	−	+
				2	−	+	+	+	+
				3	+	−	−	+	−
				4	+	+	−	−	−
4	−	−	+	1	−	+	−	+	−
				2	−	+	+	−	−
				3	+	−	−	−	+
				4	+	−	+	+	+

designs, **ILP16** and **T16**, are nonregular. The **ILP16** and **T16** designs clearly have a smaller  $G$ -aberration than the two regular designs. The design **C16** involves six  $\mathcal{SS}$  interactions that are completely aliased with the main effects of the whole-plot factors, while the design **BS16** involves two such interactions. The  $\mathcal{WSS}$  frequency vectors show that the  $\mathcal{SS}$  interactions are only partially aliased with the main effects of the whole-plot factors when the **ILP16** and **T16** designs are used. The nonregular designs thus have a clear advantage over the regular ones.

Comparing the two nonregular designs, we see that design **T16** has a better  $\mathcal{WSS}$  vector, and therefore a better total  $F_3$  vector, than design **ILP16**. Therefore, the estimates of the main effects obtained from design **T16** would be less biased than those obtained from design **ILP16** in case no interactions are included in the model. The latter design, however, has a much better  $F_4$  vector. As a matter of fact, design **T16**, which uses a resolution-4 subplot design, involves three pairs of completely aliased  $\mathcal{SS}$  interactions, whereas design **ILP16**, which uses a resolution-5 subplot design, has no such interaction pairs. As indicated by the  $\mathcal{WSSS}$  vectors, the **T16** design also involves  $\mathcal{WS}$  interactions that are completely aliased with  $\mathcal{SS}$  interactions, while the **ILP16** design does not. Design **ILP16** is therefore superior when interactions are included in the model.

For each of the benchmark designs and the **ILP16** design, Table 4 shows the number of estimable effects for several models, ranging from a main-effects model to a model including all main effects and two-factor interactions. The eight models considered all contain an intercept

Table 3: Breakdown of the confounding frequency vectors  $F_3$  and  $F_4$  for the designs **ILP16**, **BS16**, **C16** and **T16**. Entries of the  $F_3$  and  $F_4$  vectors correspond to  $J_r$  characteristics of 16 and 8.

Vector	Type	<b>ILP16</b>		<b>BS16</b>		<b>C16</b>		<b>T16</b>	
$F_3$	$SSS$	0	0	0	0	0	0	0	0
	$WSS$	0	12	2	0	6	0	0	8
	$WWW$	1	0	1	0	1	0	1	0
	Total	1	12	3	0	7	0	1	8
$F_4$	$SSSS$	0	0	1	0	1	0	1	0
	$WSSS$	0	12	4	0	0	0	2	8
	$WWSS$	0	12	2	0	6	0	0	8
	Total	0	24	7	0	7	0	3	16

and the main effects of all whole-plot and subplot factors. The models differ in the types of two-factor interactions included.

Because the four designs are all orthogonal, they perform equally well for the main-effects model. This is shown in the first row of Table 4. With only four whole plots, it is not possible to estimate  $WW$  interaction effects in addition to the whole-plot factors' main effects. That is the reason why the number of estimable effects for the second model, labeled  $WW$  in the table, is the same as for the main-effects model for each of the designs. The nonregular **ILP16** and **T16** designs both offer seven estimable  $WS$  and  $SS$  interaction effects. As a result, the **ILP16** and **T16** designs offer a larger number of estimable effects than the **BS16** design for two of the models considered and than the **C16** design for four of the models. For each model considered, the ILP design offers at least as many estimable effects as the benchmark designs.

Table 4 shows that the **ILP16** and **T16** designs perform equally well when it comes to estimating models with many interactions. We also studied models including fewer interaction effects. It turns out that, for such models, the **ILP16** design outperforms the **T16**. This is due to the fact that the **T16** design is based on a resolution-4 subplot design, whereas the **ILP16** design uses a resolution-5 subplot design. Therefore, unlike the **T16** design, the **ILP16** design allows estimation of all models involving two  $SS$  interactions.

The poor performance of the regular designs is due to the fact that these designs are inferior in terms of the  $WSS$ ,  $SSSS$  and  $WWSS$  confounding frequency vectors. That the **C16** design has the worst  $WSS$  and  $WWSS$  vectors is reflected in its having the smallest number of estimable effects for models involving  $WS$  interactions.

As a conclusion, the design characteristics given in Table 3 and 4 show that the **ILP16** design is a better design option than the regular **BS16** and the **C16** designs. It matches the **T16** design

Table 4: Number of estimable effects for eight different models when using the **ILP16**, **BS16**, **C16** and **T16** designs.

Interactions	<b>ILP16</b>	<b>BS16</b>	<b>C16</b>	<b>T16</b>
None	9	9	9	9
$WW$	9	9	9	9
$SS$	16	15	13	16
$WS$	16	16	12	16
$WW, SS$	16	15	13	16
$WW, WS$	16	16	12	16
$SS, WS$	16	16	16	16
$WW, SS, WS$	16	16	16	16

in terms of estimating models with large numbers of interactions and outperforms it in terms of estimating models with small numbers of interactions, due to its better subplot design.

### 3.2.2 32-run designs

Bingham et al. (2004) report a nine-factor regular minimum aberration FFSP design for a cheese-making experiment with eight whole plots and four subplots per whole plot. We label the design **BSS32**. The whole-plot design was a duplicated full factorial design in two factors, while the subplot design was the regular  $2_{IV}^{7-2}$  minimum aberration design.

We constructed two alternative designs labeled **ILP32i** and **ILP32ii** using the duplicated  $2^2$  design as the whole plot design. The two designs are shown in Supplementary Section A3. Design **ILP32i** uses the same subplot design as Bingham et al. (2004), whereas design **ILP32ii** is based on the nonregular seven-factor minimum  $G$ -aberration design recommended by Schoen and Mee (2012).

The designs **BSS32**, **ILP32i** and **ILP32ii** are all resolution-4 designs, so that the  $F_3$  vector is a zero vector and  $F_4$  is the only important frequency vector. Any four columns from any of the three designs lead to a  $J_4$  characteristic of 32, 16 or 0. Table 5 shows the designs'  $SSSS$ ,  $WSSS$  and  $WWSS$  frequency vectors.

The effect of using our approach rather than that of Bingham et al. (2004) can be seen best by comparing the designs **BSS32** and **ILP32i**, because both of these involve the same whole-plot and subplot designs (as a result of which these two designs have an identical  $SSSS$  vector). The value of 4 for the  $J_4$  characteristic of 32 in the  $WSSS$  vector of design **BSS32** shows that there exist 12 pairs of one  $WS$  interaction and one  $SS$  interaction that are completely aliased (each  $J_4$  characteristic of 32 corresponds with three such pairs). The zero value for the  $J_4$  characteristic of 16 indicates there is no partial aliasing between these two types of interactions.



Table 5: Breakdown of the confounding frequency vectors  $F_4$  for the designs **BSS32**, **ILP32i**, and **ILP32ii**. Entries of the vectors correspond to  $J_4$  characteristics of 32 and 16.

Type	<b>BSS32</b>		<b>ILP32i</b>		<b>ILP32ii</b>	
$SSSS$	1	0	1	0	0	4
$WSSS$	4	0	0	16	0	16
$WWSS$	1	0	1	0	1	4
Total	6	0	2	16	1	24

Table 6: Number of estimable effects for eight different models when using the designs **BSS32**, **ILP32i**, and **ILP32ii**.

Interactions	<b>BSS32</b>	<b>ILP32i</b>	<b>ILP32ii</b>
None	10	10	10
$WW$	11	11	11
$SS$	28	28	30
$WS$	22	22	22
$WW, SS$	28	28	30
$WW, WS$	23	23	23
$SS, WS$	31	32	32
$WW, SS, WS$	31	32	32

This is completely different from design **ILP32i**, which does not involve any complete aliasing of the type  $WSSS$ , but for which 48 pairs of one  $WS$  interaction and one  $SS$  interaction are partially aliased.

Since its subplot design involves partial aliasing (as shown by its  $SSSS$  vector), design **ILP32ii** involves less complete aliasing than designs **BSS32** and **ILP32i**. Design **ILP32ii** is therefore best in terms of  $G$ -aberration. Its large degree of complete aliasing makes design **BSS32** worst in terms of  $G$ -aberration.

In Table 6, we compare the number of estimable effects for the three designs for eight different models. For the first two models, the designs perform equally well because they are all orthogonal FFSP designs with a duplicated full factorial whole-plot design. For the other six models, the numbers of estimable effects vary due to the difference in the designs' aliasing patterns.

In terms of the number of estimable effects, design **ILP32i** is only slightly better than design **BSS32**. The best design in terms of estimable effects is clearly **ILP32ii**. This is entirely due to its good subplot design. We would generally recommend this option for the design problem at hand.

### 3.3 Discussion

The two illustrations in Section 3.2 and the additional ones in Supplementary Sections A1-A3 show that our one-step ILP approach yields attractive FFSP designs that compare favorably to various benchmark designs found in the literature. In addition, the one-step approach can be used to find designs for situations in which no alternatives can be found in the literature.

The results of the one-step approach were obtained using specific weights  $d_0$ ,  $e_0$  and  $f_0$  for the maximum  $d_{ij}$ ,  $e_{ij}$  and  $f_{ij}$  values, and  $d_1$ ,  $e_1$  and  $f_1$  for the sum of the  $d_{ij}$ ,  $e_{ij}$  and  $f_{ij}$  values, in the objective function. In particular, we made the former three weights four orders of magnitude larger than the latter three, hereby indicating that we wish to avoid complete confounding. To study the dependence of the FFSP produced by the one-step approach on the weights, we also studied one design problem involving a  $2^3$  full factorial whole-plot design and a  $2^5$  full factorial subplot design in much detail, where we set  $d_0 = d_1$ ,  $e_0 = e_1$  and  $f_0 = f_1$  (indicating that we only want to minimize the total amount of aliasing, independent of whether the aliasing is complete or partial),  $d_0 = 5d_1$ ,  $e_0 = 5e_1$  and  $f_0 = 5f_1$  (indicating some preference for avoiding complete aliasing), and  $d_0 > 10d_1$ ,  $e_0 > 10e_1$  and  $f_0 > 10f_1$ . In the former case, the solution produced by our one-step approach only involves complete aliasing and no partial aliasing at all. In the latter case, the FFSP design produced only involves partial aliasing, and, in the intermediate case, there is some complete and some partial aliasing. Our experience was that, for the vast majority of the weights explored, the optimal designs produced by our approach did not involve complete aliasing, but only partial aliasing.

We also performed a detailed study of the computing times, the full report of which can be found in Supplementary Section B. The optimum assignment matrix  $\mathbf{B}$  for the 16-run case in Section 3.2.1 can be found in a couple of seconds using the CPLEX solver, and a PC with a 2.93 GHz processor and a 64-bit operating system. However, for a 32-run design with three whole-plot factors, as many as 16 subplot factors and eight whole plots of size four, the CPLEX solver did not reach a definitive solution within two hours, even though the first feasible solution was produced after eight minutes. For a 48-run design with five whole-plot factors, ten subplot factors and eight whole plots of size six, the ILP approach did not return a single feasible solution in one hour. This steep increase in computing time is due to the fact that increasing the number of runs and/or the number of factors implies a larger number of decision variables in the ILP model.

To be able to handle large FFSP design problems where the one-step ILP model fails to

produce good solutions within a reasonable computing time, we propose a much faster alternative which also relies on ILP, but which involves two sequential optimization steps instead of a single one.

## 4 Two-step approach for large two-level designs

To reduce the computation time required to find good FFSP designs, we split their construction into two steps. In the first step, an ILP model is employed to partition the subplot treatments defined by subdesign  $\mathbf{S}$  into  $b$  blocks. Next, we either use a complete enumeration approach (for problems involving few whole plots) or a variable neighborhood search algorithm (for problems involving many whole plots) to assign the whole-plot factor level combinations in subdesign  $\mathbf{M}$  to the blocks. We call this approach the two-step approach.

We implemented the two-step approach in SAS and Matlab. Again, we call the CPLEX solver from within Matlab. The SAS and Matlab implementations of the two-step approach are available in the supplementary materials.

### 4.1 Step 1: Partitioning of subplot treatments in blocks

As in the one-step approach, we start with a subdesign  $\mathbf{S}$  containing  $n$  subplot treatments and a subdesign  $\mathbf{M}$  involving  $b$  whole-plot treatments. In the first step, we use an ILP model to find an optimal binary blocking matrix  $\mathbf{B}$  of dimension  $n \times b$ , so that  $\mathbf{B}$  is orthogonal to the subdesign  $\mathbf{S}$  and there is minimum confounding between the columns of  $\mathbf{B}$  and the  $\mathcal{SS}$  interaction contrasts in the  $n \times s_2$  matrix  $\mathbf{Z}$ . The blocking matrix  $\mathbf{B}$  partitions the  $n$  subplot treatments in  $\mathbf{S}$  into  $b$  blocks of size  $n/b$ . A value of 1 for an element  $b_{ij}$  of  $\mathbf{B}$  indicates that the  $i$ th row of the subplot treatment matrix  $\mathbf{S}$  is assigned to block  $j$ .

An important feature of the ILP model in the first step of our two-step procedure is that it involves fewer decision variables than the ILP model in Section 3, so that an optimal solution can be found more quickly.

Define the  $s_2 \times b$  matrix  $\mathbf{G}$  as  $\mathbf{G} = \mathbf{Z}^T \mathbf{B}$  and  $g_{ij}$  as the absolute value of the entry of  $\mathbf{G}$  in the  $i$ th row and  $j$ th column. To minimize the confounding between the blocks and the  $\mathcal{SS}$  interactions, we need to minimize the matrix  $\mathbf{G}$  in some sense. To this end, we define an objective function which is a weighted sum of (1)  $g_{max}$ , the maximum of all  $g_{ij}$  values, (2)  $g_{cmax}$ , the maximum of the  $s_2$  sums  $\sum_{j=1}^b g_{ij}$ , and (3)  $g_{sum}$ , the overall sum of all  $g_{ij}$  values. Minimizing

$g_{max}$  ensures that there is no block whose runs have a particularly low information content for any  $\mathcal{SS}$  interaction, while minimizing  $g_{cmax}$  ensures that there is no single  $\mathcal{SS}$  interaction which is severely confounded with the blocks. Minimizing the third quantity,  $g_{sum}$ , deals with the remaining confounding. The weights for  $g_{max}$ ,  $g_{cmax}$  and  $g_{sum}$  in the objective function are  $c_1 = 10^6$ ,  $c_2 = 10^4$  and  $c_3 = 1$ , respectively.

The ILP model we employed is as follows:

$$\text{Minimize } c_1 g_{max} + c_2 g_{cmax} + c_3 g_{sum}, \quad (20)$$

Subject to

Orthogonality constraints:

$$\mathbf{S}^T \mathbf{B} = \mathbf{0}, \quad (21)$$

Confounding constraints:

$$\mathbf{z}_i^T \mathbf{b}_j - g_{ij} \leq 0; \quad i = 1, \dots, s_2; j = 1, \dots, b, \quad (22)$$

$$\mathbf{z}_i^T \mathbf{b}_j + g_{ij} \geq 0; \quad i = 1, \dots, s_2; j = 1, \dots, b, \quad (23)$$

$$0 \leq g_{ij} \leq g_{max}; \quad i = 1, \dots, s_2; j = 1, \dots, b, \quad (24)$$

$$0 \leq \sum_{j=1}^b g_{ij} \leq g_{cmax}; \quad i = 1, \dots, s_2, \quad (25)$$

Technical constraints:

$$\mathbf{1}_n^T \mathbf{B} = \frac{n}{b} \mathbf{1}_b^T, \quad (26)$$

$$\mathbf{B} \mathbf{1}_b = \mathbf{1}_n; \quad (27)$$

$$b_{ij} \in \{0, 1\}; \quad i = 1, \dots, n; j = 1, \dots, b. \quad (28)$$

## 4.2 Step 2: Assignment of whole-plot treatments

In the second step of the two-step approach, we need to assign each row of the whole-plot design  $\mathbf{M}$  to one of the blocks created in the first step, i.e. to a column of the blocking matrix  $\mathbf{B}$ . We denote a specific assignment by the vector  $\mathbf{a} = (a_1, a_2, \dots, a_b)$ , which is a permutation of the values  $1, 2, \dots, b$  and where  $a_j = k$  means that the  $k$ th row of the whole-plot treatment matrix  $\mathbf{M}$  is assigned to the  $j$ th block identified in step 1.

In general, the FFSP design resulting from each permutation will have a different confounding frequency vector and a different number of estimable effects. To select the best possible permutation, we use the objective function given in (3). When the number of whole plots is limited, it is computationally feasible to enumerate all possible assignments of the whole-plot treatments to the blocks, i.e. to enumerate all  $b!$  permutations of the elements of  $\{1, 2, \dots, b\}$ , to evaluate the objective function for each permutation, and to select the best one. As soon as

the number of whole plots exceeds ten, it is, however, computationally more efficient to use a variable neighborhood search (VNS) algorithm instead of a complete enumeration.

A VNS algorithm is a variant of a local search algorithm which uses more than one neighborhood to prevent the search process from getting stuck in a locally optimal solution (Mladenović and Hansen, 1997; Hansen et al., 2008). Garroi et al. (2009) indicate that VNS approaches have been successfully applied to a variety of combinatorial optimization problems and used a VNS algorithm themselves to find the best run order for central composite designs in the presence of serial correlation.

The local search approach to find an optimal permutation involves an iterative process that improves the permutation step by step. At each step, an improved permutation is obtained by making a minor change to the current one. The minor change is defined by a specific rule, and all possible changes that can be made to a given permutation according to that rule form a set of alternative solutions called neighborhood. A member of a neighborhood is a neighbor.

One possible neighborhood for a given permutation  $\mathbf{a}$  is obtained by selecting any two elements of  $\mathbf{a}$  and swapping their positions. For example, suppose that the current solution is the vector  $\mathbf{a} = (1, 2, 3, 4, 5, 6, 7, 8)$ . Two neighbors of that solution are  $\mathbf{a}^1 = (2, 1, 3, 4, 5, 6, 7, 8)$  and  $\mathbf{a}^2 = (1, 2, 3, 4, 5, 6, 8, 7)$ . The local search algorithm either determines which neighbor has the best objective function value (in which case the algorithm is a steepest-descent or best-improvement algorithm) or finds any neighbor that has a better objective function value (in which case the algorithm is a first-improvement algorithm). The selected neighbor then replaces the current solution. The local search algorithm then continues by examining the neighbors of the new solution and stops if no better solution can be found any more.

The danger of a local search algorithm is that it may get stuck in a locally optimal solution instead of a global optimum because it does not examine all possible changes to the current solution. A VNS algorithm attempts to overcome this weakness of classical local search algorithms by using more than one neighborhood. The rationale for using more than one neighborhood is that a solution which is a local optimum with respect to one neighborhood is not necessarily a local optimum with respect to another neighborhood. A key feature of the VNS algorithm is that it allows the search to jump from one neighborhood to another when a solution is a local optimum with respect to the former neighborhood.

For the problem of finding an optimal assignment of whole-plot treatments to blocks, we have implemented a first-improvement VNS algorithm involving the three neighborhoods defined in

Table 7: Neighborhood structures used by the VNS algorithm.

Neighborhood	Size	Description	Examples
$N_1$	$O(b)$	Shift all the items one position to the left, and put the leftmost item in the rightmost position	$[1, 2, 3, 4, 5] \rightarrow [2, 3, 4, 5, 1], [3, 4, 5, 1, 2]$
$N_2$	$O(b^2)$	Swap any two items	$[1, 2, 3, 4, 5] \rightarrow [5, 2, 3, 4, 1], [1, 2, 4, 3, 5]$
$N_3$	$O(b^3)$	Choose any subset of three items, and shift each item in that subset one position to the left or to the right	$[1, 2, 3, 4, 5] \rightarrow [2, 3, 1, 4, 5], [3, 1, 2, 4, 5]$

Table 7. We opted for a first-improvement algorithm because this type of algorithm does not require evaluating the objective function value of every possible neighbor of a given solution. The sizes of the three neighborhoods  $N_1$ ,  $N_2$  and  $N_3$  increase linearly, quadratically and cubically with the number of whole plots. Especially for the larger neighborhoods, the first-improvement approach results in substantial computing time savings.

Starting from an initial random permutation, the VNS algorithm begins by exploring the smallest neighborhood, which is  $N_1$ , and identifying a better permutation in that neighborhood. As soon as such a better permutation has been found, the algorithm attempts to improve this newly obtained permutation, by exploring its  $N_1$  neighborhood. This procedure is repeated until the VNS algorithm cannot find a better permutation any more in neighborhood  $N_1$ . The algorithm then moves to the larger neighborhood  $N_2$  of the current permutation. As soon as a better permutation is found in  $N_2$ , the algorithm returns to the  $N_1$  neighborhood of the new permutation. If no better solution can be found in  $N_2$ , the algorithm starts exploring  $N_3$ . As soon as the algorithm encounters a better permutation in  $N_3$ , the algorithm returns to the  $N_1$  neighborhood. The algorithm stops if no better solution can be found in  $N_3$ . An outline of our VNS algorithm, as embedded in the two-step approach, is presented in Algorithm 1.

Like more traditional local search algorithms, and in spite of its being less prone to get stuck in a local optimum, the VNS algorithm cannot guarantee that the resulting solution is a global optimum. Therefore, we run the VNS algorithm multiple times, starting from a different random solution each time, to increase the likelihood that the global optimum is found. In the pseudocode, the total number of runs of the VNS algorithm is labeled *maxiter*, while the number of the current run is *numiter*.

```

step #1;
determine whole-plot and subplot design matrices M and S;
run the ILP model in Section 3.1 to optimize the blocking matrix B;

step #2;
set numiter  $\leftarrow$  1;
while numiter  $\leq$  maxiter do
    generate random permutation perm;
    set k  $\leftarrow$  1;
    while k  $\leq$  3 do
        find a permutation, newperm, in  $N_k$  which is better than perm with respect to the
        objective function in (3);
        if newperm is found then
            perm  $\leftarrow$  newperm;
            k  $\leftarrow$  1;
        else
            k  $\leftarrow$  k + 1;
        end
    end
    store perm;
    set numiter  $\leftarrow$  numiter + 1;
end
report the best perm as the final solution;

```

**Algorithm 1:** Two-step approach for generating large FFSP designs, including the VNS algorithm.

### 4.3 Illustration

In this section, we illustrate the two-step approach by means of a 48-run FFSP design involving five whole-plot factors ( $\mathcal{W}_1, \dots, \mathcal{W}_5$ ), ten subplot factors ( $\mathcal{S}_1, \dots, \mathcal{S}_{10}$ ) and eight whole plots of size six. In Supplementary Section A5, we present an additional illustration involving 96 runs, five whole-plot factors, 25 subplot factors and 12 whole plots of size eight. For the additional example, we also discuss the added value of the VNS approach over a random assignment of whole-plot treatments to the blocks of subplot treatments.

For the example with eight whole plots of size six, the only possible whole-plot treatment design **M** is a regular  $2^{5-2}$  design with generators  $\mathcal{W}_4 = \mathcal{W}_1\mathcal{W}_2$  and  $\mathcal{W}_5 = \mathcal{W}_1\mathcal{W}_3$ . For subplot design **S**, we use the minimum  $G$ -aberration design involving ten two-level factors in 48 runs identified by Schoen and Mee (2012).

The first step of the two-step approach is to solve the ILP problem in (20)-(28) to obtain an optimal blocking matrix **B**, which partitions the 48 runs of subdesign **S** into eight groups of size six. The optimal arrangement of **S** into eight blocks is shown in Supplementary Section A4. The next step is to assign the eight whole-plot treatments defined by the regular  $2^{5-2}$  design

to the eight blocks. Generally, different assignments lead to different values of the objective function in (3). For instance, applying the assignment vector  $(1, 2, 3, 4, 5, 6, 7, 8)$  yields a value of 243,655,668 for the objective function in (3). Using the vector  $(2, 1, 3, 4, 5, 6, 7, 8)$  instead, the objective function value drops to 243,624,672. Therefore, the latter assignment results in a better FFSP design than the former.

We used 100 iterations of our VNS algorithm to find the best assignment of the whole-plot treatments to the eight blocks of size six. The best permutation we obtained for the assignment was  $\mathbf{a}_{opt} = (3, 2, 5, 7, 8, 6, 4, 1)$ . A complete enumeration of all possible whole-plot treatment assignments in a later phase confirmed that the VNS algorithm had produced the globally optimal permutation. That permutation results in an objective function value of 164,328,096. The interpretation of the vector  $\mathbf{a}_{opt}$  is that the final 48-run FFSP design, which we label **ILP48**, can be constructed by assigning the first row of  $\mathbf{M}$  to the eighth block of subplot treatments, the second row of  $\mathbf{M}$  to the second block, etc. The design **ILP48** is shown in Supplementary Section B4.

We compare our **ILP48** design with an alternative proposed by Kulahci and Bisgaard (2005), which we label **KB48**. While Kulahci and Bisgaard (2005) used the same whole-plot design as we did, their 48-run subplot design was obtained from two repeats of a 12-run Plackett-Burman design with 11 factors and two repeats of the folded-over version of this design (while we used a 48-run minimum  $G$ -aberration design). One of the 11 subplot factors in Kulahci and Bisgaard (2005) was aliased with a  $WW$  interaction to split each of the four Plackett-Burman designs in two whole plots of size six.

Table 8 presents the confounding frequency vectors of the **ILP48** and **KB48** designs. The **ILP48** design performs better than the **KB48** design in terms of the  $G$ -aberration criterion. However, the **KB48** design has the better  $WSS$ ,  $WSSS$  and  $WWSS$  vectors. This is because the construction method of Kulahci and Bisgaard (2005) specifically aimed at estimating as many  $WS$  interactions as possible. The **ILP48** design is better for the  $SSS$  and  $SSSS$  aliasing types because we used a minimum  $G$ -aberration design for the subplot treatments. Therefore, design **ILP48** offers a larger number of estimable  $SS$  interaction effects than its alternative.

The numbers of estimable effects for various models when using the **KB48** and **ILP48** designs are shown in Table 9. By construction, the **KB48** design performs well for a model with main effects and  $WS$  interactions. However, it does not outperform the **ILP48** design, which was not constructed specifically for estimating  $WS$  interactions. The **KB48** design performs poorly



Table 8: Breakdown of the confounding frequency vectors  $F_3$  and  $F_4$  for the designs **ILP48** and **KB48**. Entries of the  $F_3$  vector correspond to  $J_3$  characteristics of 48, 16 and 8. Entries of the  $F_4$  vector correspond to  $J_4$  characteristics of 48, 32, 24, 16 and 8.

Vector	Type	<b>ILP48</b>					<b>KB48</b>				
$F_3$	$SSS$	0		0	0		0		120	0	
	$WSS$	0		24	87		0		0	0	
	$WWW$	2		0	0		2		0	0	
	Total	2		24	87		2		120	0	
$F_4$	$SSSS$	0	0	0	48	0	0	0	0	210	0
	$WSSS$	0	1	4	72	232	0	0	0	0	0
	$WWSS$	0	4	0	44	186	0	0	0	90	0
	$WWWW$	1	0	0	0	0	1	0	0	0	0
	Total	1	5	14	164	418	1	0	0	300	0

Table 9: Number of estimable effects for eight different models when using the **ILP48** and **KB48** designs.

Interactions	<b>ILP48</b>	<b>KB48</b>
None	16	16
$WW$	18	18
$SS$	45	17
$WS$	46	46
$WW, SS$	46	18
$WW, WS$	48	48
$SS, WS$	48	47
$WW, SS, WS$	48	48

for a model containing  $SS$  interactions. For three models, the **ILP48** design outperforms the **KB48** design, and it matches the **KB48** design's performance for all other models. For this reason, **ILP48** is preferable to the **KB48** design.

#### 4.4 Comparison with one-step results

One of the main motivations to develop the two-step approach was to reduce the computing time. In Supplementary Section B, we compare the computing times for the one-step and two-step approaches for 24-run, 32-run and 48-run FFSP designs, several whole-plot sizes and several numbers of factors. The results show that the computing times are substantially smaller when using the two-step approach than when using the one-step approach.

When applied to the same problem, the one-step and two-step approaches may result in different designs. In this section, we study the differences between the two approaches by means of a design problem involving a  $2^3$  full factorial whole-plot design and a  $2^5$  full factorial subplot design. The designs we obtained are given in Supplementary Section A2.

Table 10: Matrix  $\mathbf{Z}^T\mathbf{B}$ , measuring the confounding of  $\mathcal{SS}$  interactions with the eight blocks or whole plots in the one-step and two-step approach for the example in Section 4.4.

Interaction	One-step approach								Two-step approach							
$\mathcal{S}_1\mathcal{S}_2$	0	0	0	0	0	0	0	0	0	0	0	-4	4	0	0	0
$\mathcal{S}_1\mathcal{S}_3$	0	0	0	0	0	0	0	0	0	-4	0	0	0	0	4	0
$\mathcal{S}_1\mathcal{S}_4$	0	0	0	0	0	0	0	0	0	0	-4	0	0	4	0	0
$\mathcal{S}_1\mathcal{S}_5$	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	-4
$\mathcal{S}_2\mathcal{S}_3$	0	4	0	-4	0	-4	0	4	-4	0	4	0	0	0	0	0
$\mathcal{S}_2\mathcal{S}_4$	-4	0	4	0	4	0	-4	0	0	-4	0	0	0	0	0	4
$\mathcal{S}_2\mathcal{S}_5$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\mathcal{S}_3\mathcal{S}_4$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\mathcal{S}_3\mathcal{S}_5$	4	0	-4	0	-4	0	4	0	0	0	0	-4	0	4	0	0
$\mathcal{S}_4\mathcal{S}_5$	0	4	0	-4	0	-4	0	4	0	0	0	0	4	0	-4	0

In Table 10, we show the confounding pattern of the  $\mathcal{SS}$  interactions with the eight whole plots in both approaches. The confounding is described by means of a  $10 \times 8$  matrix  $\mathbf{Z}^T\mathbf{B}$ , where the rows correspond to the 10  $\mathcal{SS}$  interactions and the columns correspond to the blocks or whole plots. The elements of the matrices are the sums of a particular  $\mathcal{SS}$  interaction contrast in a particular whole plot. An important feature of the two confounding patterns is that the total amount of confounding is the same for the one-step approach and the two-step approach. This can be seen from the fact that the sum of the absolute values in the two matrices in the table is identical for the two approaches.

For the one-step approach, however, we see that six of the  $\mathcal{SS}$  interactions are not confounded with the whole plots, while there are four interactions that have absolute sums of 4 in four of the whole plots. For the two-step approach, only two interactions are not confounded with the whole plots, but the other eight interactions have absolute sums of 4 in only two of the whole plots. This is due to the first step of the two-step approach, where the term involving  $g_{cmax}$  in the objective function ensures that there is not a single  $\mathcal{SS}$  interaction which is severely confounded with the whole plots. For this reason, the confounding of the  $\mathcal{SS}$  interactions is more evenly distributed in the two-step approach than in the one-step approach. The even distribution makes it difficult for the second step of the two-step approach to avoid aliasing of the type  $\mathcal{WSS}$ . This is visible in Table 11, where we compare the breakdown of the  $F_3$  and  $F_4$  confounding frequency vectors for the one-step and two-step approaches. The table's first row shows that the one-step approach is capable of finding a split-plot arrangement for which the  $\mathcal{SS}$  interactions are not at all aliased with the main effects of the whole-plot factors. The second step of the two-step approach does not manage to match this result, due to the fact that the aliasing with the whole-plot factors'

Table 11: Breakdown of the confounding frequency vectors  $F_3$  and  $F_4$  for designs obtained by the one-step and two-step approach. The only entry of the  $F_3$  vector corresponds to a  $J_3$  characteristic of 8. The two entries of the  $F_4$  vector correspond to  $J_4$  characteristics of 16 and 8.

Type	One-step approach		Two-step approach	
$WSS$	0	0	0	8
$WWSS$	4	0	0	16
$WSSS$	8	0	4	12

main effects and interaction effects is to a large extent fixed in the approach's first step. The excellent performance of the one-step approach in terms of aliasing of the type  $WSS$ , relative to the two-step approach, is to some extent offset by a poorer performance in terms of the aliasing of the type  $WWSS$ .

To summarize the comparison between both approaches, the one-step approach prioritizes minimization of aliasing of the type  $WSS$ , following by aliasing of the type  $WWSS$ . Due to its first step, the two-step approach prioritizes the confounding of the  $SS$  interactions with the whole plots. This guarantees a limited amount of aliasing with any possible contrast concerning whole-plot factor effects, but not with any specific one. Only in its second step, the two-step approach's focus is on aliasing of the types  $WSS$  and  $WWSS$ .

In Supplementary Section C2, we report the results of a sensitivity study for the two-step approach, similar to that for the one-step approach. Compared with the one-step approach, the two-step approach is far less sensitive to the settings of the weights of  $d_0$ ,  $d_1$ ,  $e_0$ ,  $e_1$ ,  $f_0$  and  $f_1$  in the objective function, because its first step already minimizes the confounding of  $SS$  interactions with the whole plots.

## 5 Multi-level designs

Many experiments do not just involve two-level factors, but they also involve multi-level factors, which act at more than two levels. In this section, we describe how to adapt our one-step and two-step procedures to construct FFSP designs with one or more multi-level factors.

For two-level designs, the aliasing between main effects and two-factor interactions and the aliasing among two-factor interactions can be measured well using confounding frequency vectors. These confounding frequency vectors were unambiguous, because, in two-level designs, contrast coefficients can only take the values  $-1$  or  $+1$ . In multi-level designs, the confounding frequency vectors depend on the particular set of contrasts chosen. However, the sums of the squared

correlations implied do not depend on the particular contrasts chosen (Xu and Wu, 2001). These sums of squares constitute the generalized word length pattern  $(A_3, A_4, \dots, A_{w+s})$ , where an element  $A_i$  is called the generalized word count of length  $i$ . We quantify the aliasing in the multi-level FFSP design using generalized word counts of length 3 and length 4,  $A_3$  and  $A_4$ .

After detailing our approach to construct multi-level designs, we demonstrate its usefulness by revisiting an experiment on DNA amplification described in Schoen and Wolff (1997).

## 5.1 Construction procedure

As before, our approach starts by determining a suitable whole-plot design, which we now call  $\mathbf{M}^*$ , and a suitable subplot treatment design, which we now call  $\mathbf{S}^*$ . The whole-plot design involves  $b$  whole-plot treatments and  $w$  whole-plot factors,  $\mathcal{W}_1, \mathcal{W}_2, \dots, \mathcal{W}_w$ , with  $\kappa_1, \kappa_2, \dots, \kappa_w$  levels, respectively. The subplot design involves  $n$  subplot treatments and  $s$  subplot factors,  $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_s$ , with  $\lambda_1, \lambda_2, \dots, \lambda_s$  levels, respectively. The term multi-level design refers to any situation where at least one  $\kappa_i$  or  $\lambda_j$  value is greater than two. A necessary condition for the  $b \times w$  whole-plot design  $\mathbf{M}^*$  to be orthogonal is that  $b$  is a common multiple of any pairwise product  $\kappa_i \kappa_j$  (where  $i \neq j$ ). A necessary condition for the  $n \times s$  subplot design  $\mathbf{S}^*$  to be orthogonal is that  $n$  is a common multiple of any pairwise product  $\lambda_i \lambda_j$  (where  $i \neq j$ ). Finally, because we require that the blocking matrix  $\mathbf{B}$  be orthogonal to  $\mathbf{S}^*$ , the whole-plot size  $n/b$  has to be a common multiple of each  $\lambda_i$ .

In the case of two-level FFSP designs (where all  $\kappa_i$  and  $\lambda_j$  equal 2 for all  $i$  and  $j$ ),  $n$  and  $b$  must be multiples of four and  $n/b$  has to be even. If a three-level FFSP design is desired (where all  $\kappa_i$  and  $\lambda_j$  are 3),  $n$  and  $b$  must be multiples of nine and  $n/b$  has to be a multiple of three. These requirements are more complex when a mixed-level FFSP design (where not all  $\kappa_i$  and  $\lambda_j$  are equal) is constructed. For example, suppose that a FFSP design with one three-level and one two-level whole-plot factor and four two-level subplot factors is desired. Then,  $b$  must be a multiple of six,  $n$  must be a common multiple of six and four, and  $n/b$  has to be even. As a result, the smallest design options are a 12-run design in six whole plots, a 24-run design in 12 whole plots, or a 24-run design in six whole plots.

To construct multi-level FFSP designs, we propose the following approach:

1. Determine a  $b \times w$  orthogonal design  $\mathbf{M}^*$  for the whole-plot factors and an  $n \times s$  orthogonal design  $\mathbf{S}^*$  for the subplot factors.
2. For each  $i \in \{1, \dots, w\}$ , convert the  $i$ th column of  $\mathbf{M}^*$ , corresponding to the  $i$ th whole-

- plot factor, into  $(\kappa_i - 1)$  orthogonal contrast columns. The result is a matrix  $\mathbf{M}$  with  $w_1 = \sum_{i=1}^w \kappa_i - w$  columns.
3. Generate the contrast matrix  $\mathbf{V}$  for the  $\mathcal{WW}$  interactions, by calculating the elementwise product of each of the  $w_2 = \sum_{i=1}^w \sum_{j=i+1}^w (\kappa_i - 1)(\kappa_j - 1)$  pairs of columns of  $\mathbf{M}$  that correspond to different factors. Note that the interaction of two whole-plot factors with  $\kappa_i$  and  $\kappa_j$  levels involves  $(\kappa_i - 1)(\kappa_j - 1)$  interaction components.
  4. For each  $i \in \{1, \dots, s\}$ , convert the  $i$ th column of  $\mathbf{S}^*$ , corresponding to the  $i$ th subplot factor, into  $(\lambda_i - 1)$  orthogonal contrast columns. The result is a matrix  $\mathbf{S}$  with  $s_1 = \sum_{i=1}^s \lambda_i - s$  columns.
  5. Generate the contrast matrix  $\mathbf{Z}$  for the  $\mathcal{SS}$  interactions, by calculating the elementwise product of each of the  $s_2 = \sum_{i=1}^s \sum_{j=i+1}^s (\lambda_i - 1)(\lambda_j - 1)$  pairs of columns of  $\mathbf{S}$  that correspond to different factors. Note that the interaction of two subplot factors with  $\lambda_i$  and  $\lambda_j$  levels involves  $(\lambda_i - 1)(\lambda_j - 1)$  interaction components.
  6. Normalize all columns of  $\mathbf{M}$  and  $\mathbf{V}$ , so that they have a squared norm of  $b$ .
  7. Normalize all columns of  $\mathbf{S}$  and  $\mathbf{Z}$ , so that they have a squared norm of  $n$ .
  8. Obtain a FFSP design using either the one-step approach in Section 3, or for, larger problems, the two-step approach in Section 4.

The normalization of the matrices  $\mathbf{M}$ ,  $\mathbf{V}$ ,  $\mathbf{S}$  and  $\mathbf{Z}$  results in an equal weight for each of the contrasts when minimizing the aliasing. Except for a pure two-level FFSP design, the normalization results in non-integer contrast matrices. Therefore, the linear programming model for multi-level designs is a mixed integer linear programming (MILP) model, which has non-integer decision variables in addition to the binary variables contained within  $\mathbf{B}$ .

## 5.2 Illustrations

In Supplementary Section A6, we use our two-step approach to construct a three-level FFSP design with 81 runs, based on the strength-three design labeled 81.8.234 in Sartono et al. (2012) and involving two whole-plot factors, eight subplot factors and nine whole plots of size nine. In this section, we focus on a real-life 48-run experiment on DNA amplification, involving one three-level and one two-level whole-plot factor, one four-level subplot factor, four two-level subplot factors, and 12 whole plots of size four (Schoen and Wolff, 1997). The challenge here is to

combine a 48-run subplot design involving one four-level factor and four two-level factors with a 12-run whole-plot design, which is a duplicated  $3 \times 2$  full factorial design. There are four non-isomorphic orthogonal arrays of strength 3 that can serve as a subplot design for this example (Schoen et al., 2010). The design we used is the design with the smallest generalized word count of length 4.

Following steps 2 and 4 of the procedure outlined in Section 5.1, we created three orthogonal main-effect contrast vectors for the four-level factor by replacing the four factor levels by the vectors  $(-1, -1, 1)$ ,  $(1, -1, -1)$ ,  $(-1, 1, -1)$  and  $(1, 1, 1)$ . For the three-level factor we created two orthogonal main-effect contrast vectors by replacing the first level of each factor by  $(2, 0)$ , the second level by  $(-1, 1)$  and the third level by  $(-1, -1)$ .

After normalizing the contrast columns, we used the two-step approach discussed in Section 4 to find the best possible FFSP design. The design, which we refer to as **MILP48**, is presented in Table 12.

The design of Schoen and Wolff (1997), which we label **SW48**, was based on a regular resolution-4  $2^{8-3}$  FFSP design with eight whole plots of size four. Initially, there were two two-level whole-plot factors, one of which indicating the DNA amplification device to be used. Each device was used for four whole plots. Just before the start of the experiment, a third device became available. The experimenters decided to copy the four whole plots for one of the original devices and use them for the third device as well. This resulted in 12 whole plots instead of eight. The final design did therefore not have orthogonal main-effect contrast vectors, unlike our **MILP48** design. Moreover, and more importantly, the **SW48** design did not permit estimation of all main effects and two-factor interactions, due to its resolution of 4. The design's defining relation involves five words of length 4, each of which gives rise to three pairs of two-factor interactions which are not jointly estimable. Schoen and Wolff (1997) discuss several other design options with less severe aliasing, but all of these involve regular designs and result in non-estimable two-factor interactions.

The aliasing in the **MILP48** design is shown in Table 13. Clearly, the most serious aliasing occurs between  $WS$  interactions and  $SS$  interactions. This is not a major weakness, however, because the design permits estimation of all main effects and all two-factor interactions. The **MILP48** design therefore outperforms the design used by Schoen and Wolff (1997), and all other designs considered by these authors.

Table 12: 48-run mixed level FFSP design involving one four-level factor, one three-level factor, and five two-level factors, labeled **MILP48**.

Whole plot	$\mathcal{W}_1$	$\mathcal{W}_2$	Sub-plot	$\mathcal{S}_1 - \mathcal{S}_5$					Whole plot	$\mathcal{W}_1$	$\mathcal{W}_2$	Sub-plot	$\mathcal{S}_1 - \mathcal{S}_5$				
1	2	0	1	0	0	0	0	0	7	0	0	1	3	1	1	1	1
			2	3	0	1	1	0				2	1	0	0	1	0
			3	1	1	0	1	1				3	2	0	1	0	0
			4	2	1	1	0	1				4	0	1	0	0	1
2	0	1	1	0	0	0	0	0	8	1	0	1	2	0	0	0	1
			2	3	1	1	0	1				2	0	1	0	1	0
			3	2	1	0	1	0				3	1	1	1	1	1
			4	1	0	1	1	1				4	3	0	1	0	0
3	2	0	1	0	0	0	1	1	9	0	0	1	3	0	1	0	1
			2	2	1	0	0	0				2	1	0	0	0	0
			3	3	1	1	1	0				3	2	1	1	1	0
			4	1	0	1	0	1				4	0	1	0	1	1
4	1	0	1	3	1	0	1	1	10	1	1	1	2	0	1	1	1
			2	0	0	1	0	1				2	1	1	0	1	0
			3	1	1	1	0	0				3	3	0	0	0	1
			4	2	0	0	1	0				4	0	1	1	0	0
5	1	1	1	2	1	1	1	1	11	2	1	1	0	1	1	0	1
			2	0	0	1	1	0				2	1	0	1	1	0
			3	1	0	0	0	1				3	3	1	0	0	0
			4	3	1	0	0	0				4	2	0	0	1	1
6	0	1	1	1	1	1	0	0	12	2	1	1	2	0	1	0	0
			2	3	0	0	1	0				2	1	1	0	0	1
			3	0	0	1	1	1				3	0	1	1	1	0
			4	2	1	0	0	1				4	3	0	0	1	1

Table 13: Breakdown of the generalized word counts of lengths 3 and 4,  $A_3$  and  $A_4$ , for the **MILP48** design.

Word type	Count
$SSS$	0
$\mathcal{W}SS$	1.04
$\mathcal{W}\mathcal{W}\mathcal{W}$	-
Total	1.04
$SSSS$	0.44
$\mathcal{W}SSS$	1.99
$\mathcal{W}\mathcal{W}SS$	0.60
$\mathcal{W}\mathcal{W}\mathcal{W}\mathcal{W}$	-
Total	3.03

## 6 Discussion

In this paper, we presented a general method to construct split-plot designs based on integer linear programming (for two-level designs) and mixed integer linear programming (for multi-level and mixed-level designs). We proposed a one-step approach for small designs and a two-step approach for large designs. The one-step approach links the various subplot treatments directly to the whole-plot treatments. The two-step approach first arranges the subplot treatments in blocks and then links the blocks to whole-plot treatments.

An important novel feature of both approaches is that they can handle experimental factors with any numbers of levels. We compared our designs with benchmark designs from the literature to demonstrate the excellent performance of our method. The newly constructed two-level, multi-level and mixed-level designs generally outperform alternative designs in the literature in terms of the number of estimable effects and in terms of the confounding frequency vectors.

In the case of multi-level designs, different types of orthogonal contrasts for the main effects may in theory lead to different results. However, our computational experience suggests that the effect of using different types of contrast vectors on the quality of the final design is minor.

We suggest several subjects that merit further research. First, the quality of the final split-plot design produced by both our one-step and our two-step approach depends on the set of whole-plot treatments and the set of subplot treatments. We recommend using whole-plot and subplot designs that have minimum generalized aberration, and refer to Deng and Tang (1999), Loepky et al. (2007), Sartono et al. (2012), Schoen and Mee (2012) and Schoen et al. (2013) for interesting two-level, three-level and mixed-level designs, but it would be useful to study the sensitivity of the design produced by our one-step and two-step approaches to the design utilized for the whole-plot and subplot treatments in more detail. Second, as one of the referees suggested, it might be possible to optimize the designs for the whole-plot treatments and the subplot treatments, along with the assignment matrix  $\mathbf{B}$ . The required optimization would obviously be more complex, but it would certainly be a useful avenue for future research. A third interesting subject for further research is to investigate the use of different objective functions and different weights for  $d_0$ ,  $d_1$ ,  $e_0$ ,  $e_1$ ,  $f_0$  and  $f_1$  in the one-step and two-step approach to deal with scenarios in which a researcher is interested in specific interaction effects.



## Supplementary materials

**Additional Results.pdf:** Section A provides a detailed discussion of two examples involving 24 and 32 runs mentioned in Table 1. The section also discusses the construction of a 96-run two-level design involving 30 factors, and an 81-run design involving 10 three-level factors, and it includes various design tables. Section B presents a computing time study for the one-step and two-step approaches based on two-level experiments with 24–48 runs, 4 or 8 whole plots, 2 or 3 whole-plot factors and various numbers of subplot factors. Section C reports the results of a study of the sensitivity of the designs produced by the one-step and two-step approaches to the weights of  $d_0$ ,  $d_1$ ,  $e_0$ ,  $e_1$ ,  $f_0$  and  $f_1$  in the objective function.

**Programs:** SAS and Matlab programs for the one-step approach and the two-step approach.

## Acknowledgments

The research of the first and third author was supported by the Flemish Fund for Scientific Research FWO. The authors are grateful to Nha Vo-Thanh for the MATLAB codes he developed and for his help when preparing a revised version of this paper. The authors are also grateful to two referees, an associate editor and the previous editor for their constructive suggestions.

## References

- Anbari, F. T. and Lucas, J. M. (2008). Designing and running super-efficient experiments: Optimum blocking with one hard-to-change factor. *Journal of Quality Technology*, 40:31–45.
- Bingham, D. R., Schoen, E. D., and Sitter, R. R. (2004). Designing fractional factorial split-plot experiments with few whole-plot factors. *Journal of the Royal Statistical Society Series C (Applied Statistics)*, 53:325–339. Corrigendum, 54:955–958.
- Bingham, D. R. and Sitter, R. R. (1999). Minimum-aberration two-level fractional factorial split-plot designs. *Technometrics*, 41:62–70.
- Capehart, S. R., Keha, A., Kulahci, M., and Montgomery, D. C. (2011). Designing fractional factorial split-plot experiments using integer programming. *International Journal of Experimental Design and Process Optimisation*, 2:34–57.
- Cheng, C. S. and Tsai, P. W. (2009). Optimal two-level regular fractional block and split-plot designs. *Biometrika*, 96:83–93.

- Deng, L. Y. and Tang, B. (1999). Generalized resolution and minimum aberration criteria for Plackett-Burman and other nonregular factorial designs. *Statistica Sinica*, 9:1071–1082.
- Deng, L. Y. and Tang, B. (2002). Design selection and classification for Hadamard matrices using generalized minimum aberration criteria. *Technometrics*, 44:173–184.
- Garroi, J. J., Goos, P., and Sørensen, K. (2009). A variable-neighbourhood search algorithm for finding optimal run orders in the presence of serial correlation. *Journal of Statistical Planning and Inference*, 139:30–44.
- Hamada, M. and Wu, C. F. J. (1992). Analysis of designed experiments with complex aliasing. *Journal of Quality Technology*, 24:130–137.
- Hansen, P., Mladenović, N., and Pérez, J. A. M. (2008). Variable neighborhood search: Methods and applications. *4OR*, 6:309–360.
- Huang, P., Chen, D., and Voelkel, J. O. (1998). Minimum-aberration two-level split-plot designs. *Technometrics*, 40:314–326.
- Jones, B. and Goos, P. (2007). A candidate-set-free algorithm for generating D-optimal split-plot designs. *Journal of the Royal Statistical Society Series C (Applied Statistics)*, 56:347–364.
- Jones, B. and Nachtsheim, C. J. (2009). Split-plot designs: What, why, and how. *Journal of Quality Technology*, 41:340–361.
- Kowalski, S. (2002). 24 run split-plot experiments for robust parameter designs. *Journal of Quality Technology*, 34:399–410.
- Kulahci, M. and Bisgaard, S. (2005). The use of Plackett-Burman designs to construct split-plot designs. *Technometrics*, 47:495–501.
- Loeppky, J. L., Sitter, R. R., and Tang, B. (2007). Nonregular designs with desirable projection properties. *Technometrics*, 49:454–467.
- Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers and Operations Research*, 24:1097–1100.
- Montgomery, D. C. (2004). *Design and Analysis of Experiments*. 6th ed., John Wiley & Sons, Inc., New York.
- Sartono, B., Goos, P., and Schoen, E. D. (2012). Classification of three-level strength-3 arrays. *Journal of Statistical Planning and Inference*, 142:794–809.

- Schoen, E. D. (1999). Designing fractional two-level experiments with nested error structures. *Journal of Applied Statistics*, 26:495–508.
- Schoen, E. D., Eendebak, P. T., and Nguyen, V. M. (2010). Complete enumeration of pure-level and mixed-level orthogonal arrays. *Journal of Combinatorial Designs*, 18:123–140.
- Schoen, E. D. and Mee, R. W. (2012). Two-level designs of strength 3 and up to 48 runs. *Journal of the Royal Statistical Society Series C (Applied Statistics)*, 61:163–174.
- Schoen, E. D., Sartono, B., and Goos, P. (2013). Optimal blocking for general resolution-3 designs. *Journal of Quality Technology*, 45:166–187.
- Schoen, E. D. and Wolff, K. (1997). Design and analysis of a fractional  $4^1 3^1 2^5$  split-plot experiment. *Journal of Applied Statistics*, 24:409 – 419.
- Tichon, J. G., Li, W., and Mcleod, R. G. (2012). Generalized minimum aberration two-level split-plot designs. *Journal of Statistical Planning and Inference*, 142:1407–1414.
- Trinca, L. A. and Gilmour, S. G. (2001). Multistratum response surface designs. *Technometrics*, 43:25–33.
- Xu, H. and Wu, C. F. J. (2001). Generalized minimum aberration for asymmetrical fractional factorial designs. *Annals of Statistics*, 29:1066–1077.